

Pregled indeksov

Iztok Savnik, FAMNIT

Prosojnice & učbenik

- Učbenik:
 - Raghu Ramakrishnan, Johannes Gehrke, *Database Management Systems, McGraw-Hill, 3rd ed., 2007.*
- *Prosojnice:*
 - *From „Cow Book“: R.Ramakrishnan,*
<http://pages.cs.wisc.edu/~dbbook/>

Podatki na pomnilniških medijih

- Diski: Direktnen dostop do blokov.
 - Branje večih zaporednih strani je cenejše kot v branje naključnih strani.
- Trakovi: Sekvenčni dostop do strani.
 - Cenejši medij kot diski; uporablja se za arhiviranje podatkov.
- Datotečna organizacija: Organizacija podatkov na pomnilniških medijih.
 - **ID zapisa (rid)** služi za identifikacijo fizične lokacije zapisa
 - **Indeksi** so podatkovne strukture, ki omogočajo iskanje ID-jev zapisov na osnovi podanih vrednosti atributov **ključa indeksa**.
- Arhitektura: **Vmesni pomnilnik** prenaša strani iz zunanjega pomnilniškega medija v bazen strani. Nivo datotek in indeksov dela z vmesnim pomnilnikom.

Alternativne datotečne organizacije

Obstaja več alternativ, *vsaka je idealna za določene situacije in ne tako dobra v drugih primerih*:

- **Neurejene datoteke**: Primerne, ko je tipični dostop branje celotne datoteke.
- **Sortirane datoteke**: Najboljše v primer, da iščemo zapise v določenem vrstnem redu ali želimo preiskati določen interval celotne množice zapisov.
- **Indeksi**: Podatkovna strukture, ki organizirajo zapise na osnovi dreves in razpršilnih funkcij.
 - Pohitrijo iskanje zapisa z dano vrednostjo iskalnega ključa.
 - Podobno sortiranim datotekam, pohitrijo iskanje podmnožice zapisov na osnovi določenih iskalnih ključev.
 - Popravljanje zapisov je mnogo hitrejše kot v primeru sortiranih datotek.

Indeksi

- Indeks pohitri iskanje zapisov na osnovi polj **iskalnih ključev** indeksa.
 - Vsaka podmnožica polj relacije je lahko iskalni ključ indeksa na dani relaciji.
 - Iskalni ključ ni enak ključu (minimalni množici polj, ki enolično določa zapis v relaciji).
- Indeks vsebuje množico **podatkovnih vpisov**
 - Podatkovni vpis: $k^* = k + rid$
- Indeks podpira učinkovito iskanje podatkovnih vpisov **k^*** z dano vrednostjo ključa **k** .
 - Za dan ključ **k** lahko poiščemo podatkovni vpis **k^*** z enim dostopom do diska.

Podatkovni vpis k^*

- Podatkovni vpis k^* je lahko:
 - Podatkovni zapis z vrednostjo ključa k .
 - $\langle k, \text{rid} \rangle$
 - $\langle k, \text{seznam rid} \rangle$
- Izbira **alternative za podatkovne vpise** je ortogonalna izbiri indeksne tehnike uporabljene za iskanje podatkovnih vpisov z vrednostjo ključa k
 - Primeri indeksnih tehnik: B+ drevesa, razpršilni indeksi

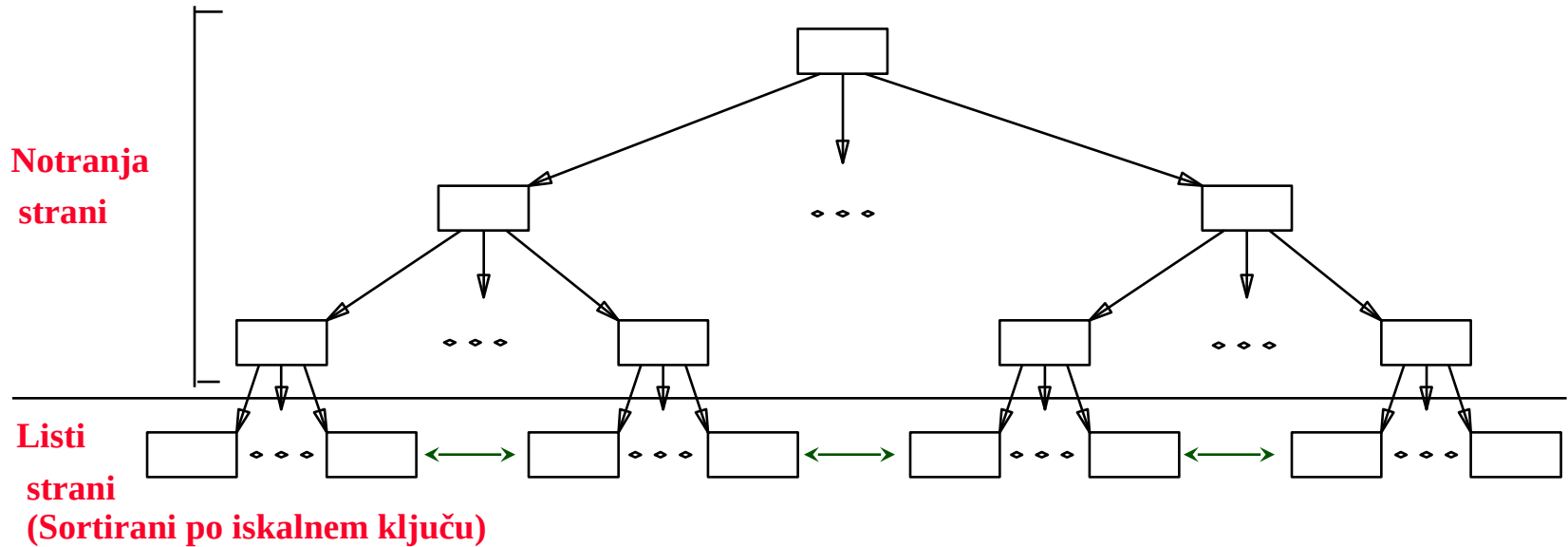
Podatkovni vpis - možnosti

- **Možnost 1:**
 - V tem primeru je indeksna struktura *datotečna organizacija*.
 - Alternativa neurejeni datoteki ali sortirani datoteki. .
 - Samo en indeks nad dano zbirko podatkov lahko uporabi možnost 1.
 - Sicer imamo podvojene podatke, ki predstavljajo redundanco ter možnost nekonsistence.
 - V primeru, da podatkovni zapisi zasedejo veliko prostora je število strani, ki vsebuje podatke veliko.
 - Tipično je tudi količina dodatnih podatkov večja.

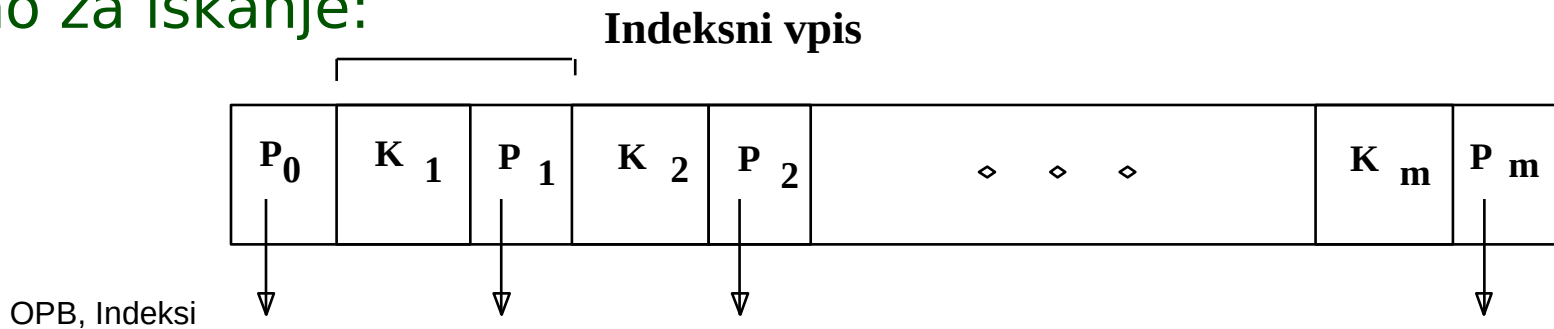
Možnosti za podatkovne vpise

- **Možnosti 2 in 3:**
 - Podatkovni vpisi so tipično precej manjši kot podatkovni zapisi.
 - Boljše od možnosti 1 v primeru velikih podatkovnih zapisov, še posebej če so iskalni ključi majhni (del indeksne strukture, ki je namenjen iskanju je precej manjši kot pri alternativni 1.).
 - Možnost 3 je bolj optimalna glede prostora od možnosti 2, vendar so zapisi variabilne dolžine tudi v primeru, da so iskalni ključi fiksne dolžine.

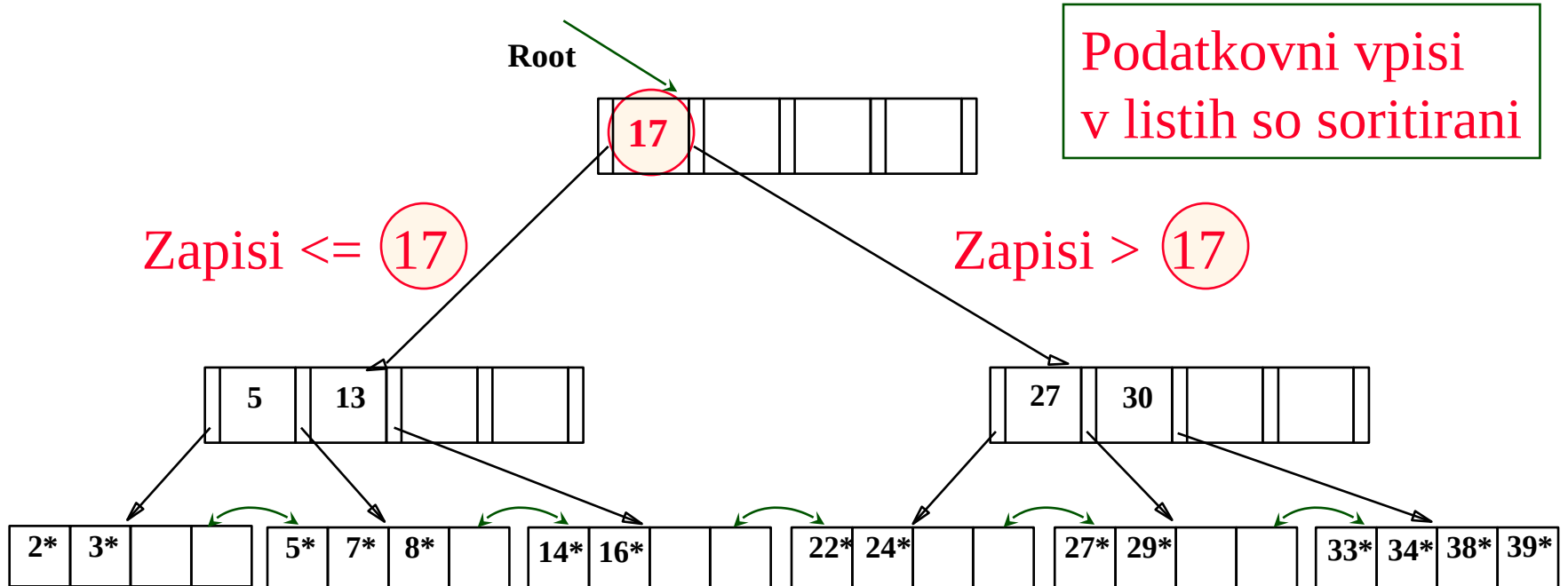
B+ Drevo



- ❖ Listi vsebujejo *podatkovne vpise*, ki so urejeni (prev & next)
- ❖ Notranje strani imajo *indeksne vpise (zapise)*; samo za iskanje:



Primer B+ Drevo



- Poišči 28*? 29*? Vse > 15* in < 30*
- Insert/delete:
 - Poišči podatkovni vpis v listu in ga spremeni.
 - Včasih se pojavi potreba po restrukturiranju staršev...
 - ... spreminjanje vozlišč "nad" spremenjenim zapisom (zbrisanim/vstavljenim).

Razpršilni indeksi

- Dobra rešitev za selekcijo z enakostjo.
- Indeks je zbirka skupkov.
 - Skupek = *primarna stran* plus nič ali več *prelivnih strani*.
 - Skupek vsebuje podatkovne vpise.
- *Razpršilna funkcija h*: $h(r)$ = skupek k kateremu pripada podatkovni zapis r . h uporabi polja *iskalnega ključa* relacije r .
 - *Ni potrebe po "indeksnih vpisih" v tej shemi.*

Klasifikacija indeksov

- *Primarni vs. Sekundarni:*
 - Če iskalni ključ vsebuje primarni ključ potem imenujemo indeks *primarni*.
 - *Unique* indeks: Iskalni ključ je kandidatni ključ.
- *Povezan vs. Nepovezan:*
 - Če je urejenost podatkovnih zapisov enaka ali blizu urejenosti podatkovnim vpisom potem pravimo, da je indeks *povezan*.
 - Možnost 1 pogojuje povezan indeks; v praksi tudi obratno: povezan indeks pogojuje možnost 1 (sortirane datoteke so redke).
 - Datoteka je lahko povezana z indeksom po samo enem ključu.
 - Cena branja podatkovnega zapisa preko indeksa zelo varira v odvisnosti ali je indeks povezan ali ne!

Povezan vs. Nepovezan indeks

- Recimo, da je možnost 2 uporabljena za podatkovne vpise in so pod. zapisi shranjeni v neurejeni datoteki.
 - Gradnja povezanega indeksa: najprej se sortira neurejena datoteka (prazen prostori za bodoče zapise).
 - Možna je uporaba so prelivnih strani za vstavljanje zapisov. V tem primeru je urejenost podatkov "blizu" prave urejenosti.

